Lecture #26 — crash consistency problem & solutions.

(*) recap:

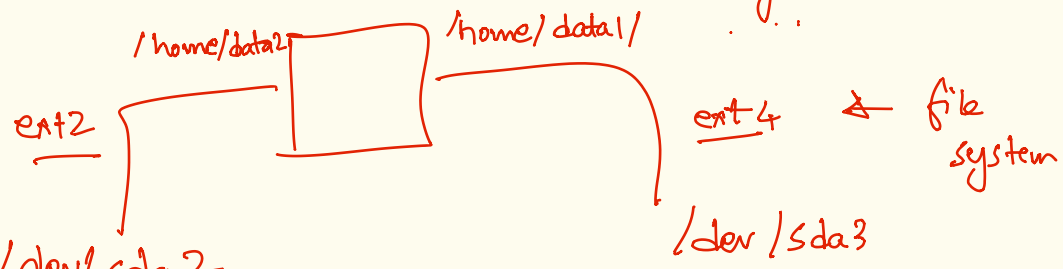
VFS — virtual file system (Linux lingo)

- standard file model
  - inode
  - superblock
  - dentry
  - file object

  } in-memory objects for file system interface and usage in OS.

- f_ops (file operations) — pointers to: read, write, open, close, seek

- each FS has to provide / adhere to file model for OS to use/operate the VFS
- individual file systems will have their own on-disk objects and corresponding operations & logic.



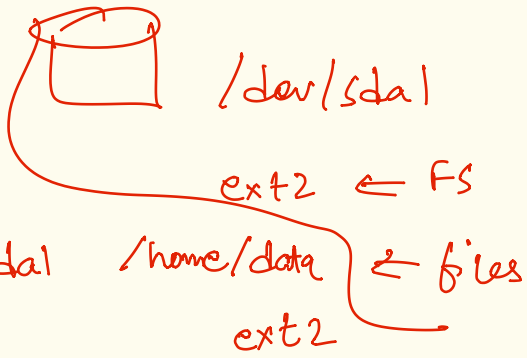
filename which points to a device

⇒ cp /home/data1/peace.txt /home/data2

- copy across file systems!

— mount ?

/home

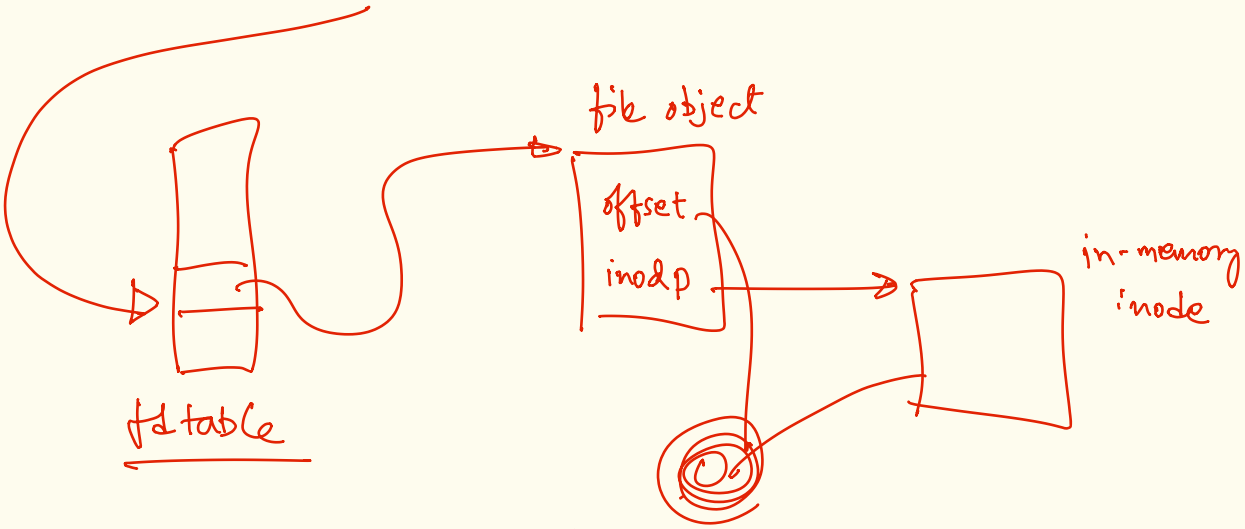


/home/data/ ↳ maps the root inode on /dev/sda1

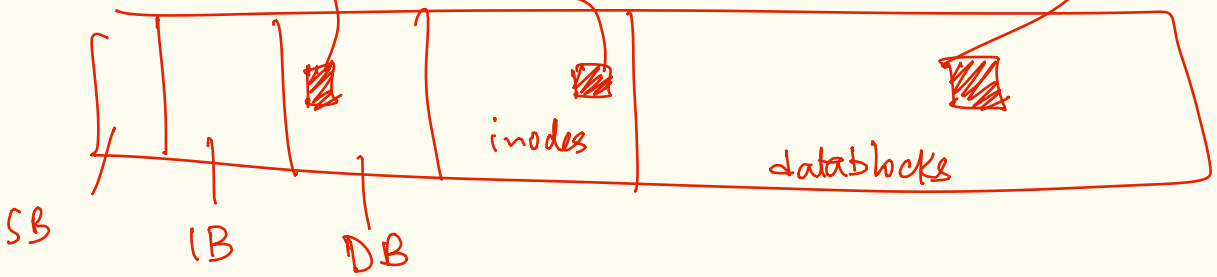
ls /home/data/

crash consistency problem.

write (fd, buf, N);



- new datablock reserved
- write to the datablock
- write/update the data bitmap
- write on the ondisk inode



(*) crash scen

crash scenarios write to a file ⇒ 3 writes to disk

C: consistent
IC: inconsistent

		inode	data bitmap	data block	
C	—	✓	✓	✓	
C	—	X	X	X	
middle IC	data lost	X	X	✓	
IC	inode has bad info.	✓	X	X	
IC	datablock leak	X	✓	X	
IC	garbage data.	✓	✓	X	!!
IC	data bitmap / block contention	✓	X	✓	*
	leak / loose datablocks data	X	✓	✓	

① fsck : file system consistency check.

- reactive process.
- given a device & the file system on it, checks for consistency violations by reading the disk.

- superblock : FS size ≤ device size
  checksum integrity check on the SB
  consistency of replicas.
- data blocks : are all datablocks accounted for
  leaks - some inode pts to the block.
  - if no pts. block bitmap can be flipped.
- inode state : all inodes in use have inode bitmap set.
- links : ref cnt. in inode matches
  # files/dentries that point to inode.

- duplicates : blocks shared in inodes.
  (need check for explicit setup or ~~viol~~ violation).

- bad blocks : out of range constants.

- directory checks — loops.

  - . .. are first 2 entries.

(2) write-ahead logging / journaling.

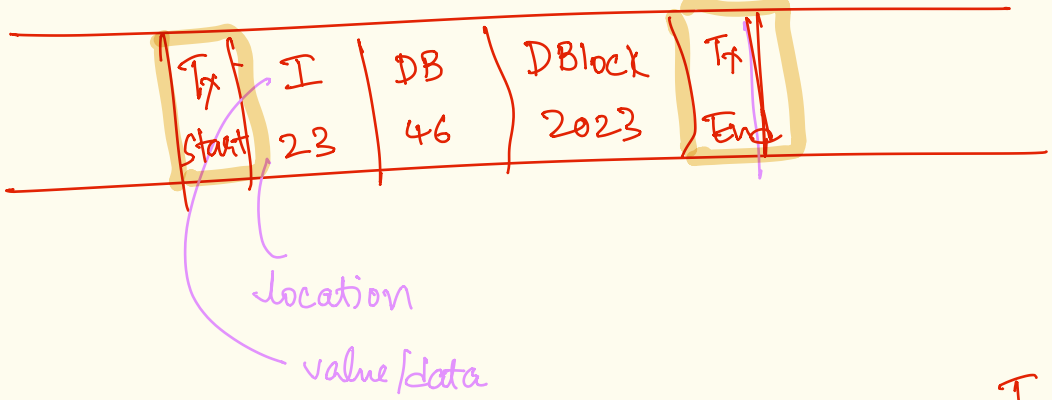
- information ~~about~~ about updates is written / logged separately.

- the log has transaction semantics.
  └ all or nothing

- log.

write

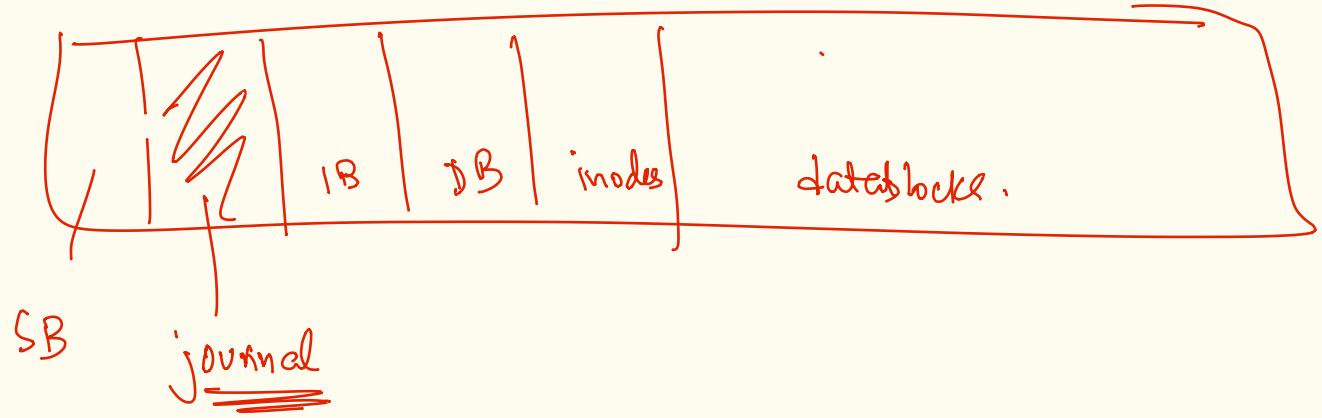
I, DB, DBlock
23 46 2023



- on (every) write.

  Tx start — updates — Tx end

  - (i) update/write log to the journal
  - (ii) commit — end of transaction.
  - (iii) Checkpoint — flush journal updates to disk (on actual locations)
  - (iii) free/reuse the log.



(3) RAID - Redundant Array of Inexpensive Independent Disks.